2 marks questions

1) Explain DevOps?

The DevOps is a combination of two words one is software Development, and second is Operations. It allows a single team to handle the entire application lifecycle, from development to **testing, deployment,** and **operations**. DevOps helps you to reduce the disconnection between software developers, quality assurance (QA) engineers, and system administrators.

2) What are the principles of DevOps?

The principles behind DevOps are:

- Continuous deployment
- Infrastructure as code
- Automation
- Monitoring
- Security

3) How DevOps is helpful to developers?

DevOps is very helpful for developers to fix the bugs and quickly implement the new features. It also helps in more transparent communication between the team members.

4) Explain some popular tools of DevOps?

Here are some popular tools of DevOps, such as:

1. **Jenkins:** Jenkins is a DevOps tool for monitoring the execution of repeated tasks. Jenkins is a software that allows continuous integration. And it will be installed on a server where the central build will take place.
2. **Ansible:** Ansible is a leading DevOps tool. Ansible is an open-source IT engine that automates application deployment, cloud provisioning, intra service orchestration, and other IT tools.
3. **Nagios:** Nagios is one of the more useful tools for DevOps. It can determine the errors and rectify them with the help of network, infrastructure, server, and log monitoring systems.
4. **Docker:** Docker is a high-end DevOps tool that allows building, ship, and run distributed applications on multiple systems.
5. **Git:** Git is an **open-source distributed version control system** that is freely available for everyone. It is designed to handle minor to major projects with speed and efficiency.

5) What are the prerequisites for the DevOps implementation?

Following are some useful prerequisites for DevOps implementation:

- Proper communication between the team members.

- At least one version control software.
- Automated testing.
- Automated deployment.

6) What are the core operation of DevOps with application development and infrastructure?

The core operation of DevOps with application development and infrastructure are:

Application development

- Code building
- Code coverage
- Unit testing
- Packaging
- Deployment

Infrastructure

- Provisioning
- Configuration
- Orchestration
- Deployment

7) What are the key components of DevOps?

The most important key components of DevOps are:

- Continuous integration
- Continuous testing
- Continuous delivery
- Continuous mongering

8) What is Vagrant?

A Vagrant is a tool of DevOps that can create and manage virtualized environments for testing and developing software.

9) What are the technical and business benefits of DevOps work culture?

Following are the technical and business benefits of DevOps work culture:

Technical benefits

- Less complex problems to fix.
- Continuous software delivery.
- Faster bugs resolution.

Business benefits

- Most stable operating environments.
- More time available to add product value.
- Faster delivery of features for customer satisfaction.

10) What is Scrum?

Scrum is used to divide a sophisticated software and product development task into smaller chunks using iterations and increasing practices. Scrum consist of three roles, such as:

1. Product owner
2. Scrum master
3. Team

11) What is version control?

It is a system that records the changes to a file or set of files over time so you can recall specific versions.

The version control system consists of a central shared repository where teammates can commit the changes to a file or set of files. It allows:

- To revert the file to the previous state.
- To compare changes over time.
- To revert the entire project to a previous state.
- To see who modified something in the files.

12) Which scripting tools are used in DevOps?

Both Python and Ruby scripting tools are used in the DevOps.

13) What is the role of AWS in DevOps?

AWS is a cloud-based service provided by Amazon that ensures scalability through unlimited computing power and storage. It empowers IT enterprises to develop and deliver experienced products and deploy applications on the cloud.

14) Why open source tools boost DevOps?

Open source tools mainly used by any organization which is adapted by DevOps pipeline because DevOps came with the focus of automation in various aspects of organization build, release, change management, and infrastructure management areas.

15) What is the need for organization playbooks as the role?

Organization playbooks as the role give more reliability and reusability to any plays while considering a task where MySQL installation should be done after the removal of Oracle DB. You

need another requirement to install MySQL after java installation. In both cases, you need to install MySQL, but without roles, it needs to write playbooks separately for both use cases. Still, once using roles, the MySQL installation role is created can be utilized any number of times by invoking using logic in **site.yaml**.

16) How many cloud platform which is used for DevOps implementation?

Here are some cloud computing platform used for DevOps implementation, such as:

- Google Cloud
- Amazon Web Services
- Microsoft Azure

17) Explain two-factor authentication?

The two-factor authentication is a security method in which the user provides two ways of identification from separate categories.

18) What is the use of PTR in DNS?

PTR is a pointer record that is used for a reverse DNS lookup.

19) Name some network monitoring tools?

Some most essential network monitoring tools are:

- Nagios
- OpenNMS
- Splunk
- Icinga 2
- Wireshark

20) Name three important DevOps KPIs?

Three most important DevOps KPIs are:

- Deployment frequency
- Percentage of failed deployments
- Mean time to failure recovery

21) Explain pair programming regarding DevOps?

Pair programming is an engineering practice of Extreme Programming Rules. In this case, two programmers work on the same system on the same design or algorithm.

One programmer act as a **Driver** and other as an **Observer** who continuously monitors the progress of a project to identify problems. The roles can be reversed at any point of time without any prior intimation.

22) What is CBD?

Component-Based Development (CBD) is a unique way to approach product development. In this method, developers always look for existing well defined, tested, and verified components to compose and assemble them to a product instead of developing from scratch.

23) Define DevOps Automation?

Automation is the crucial need for DevOps practices, and automate everything is the fundamental principle of DevOps. Automation kick starts from the code generation on the developer's machine, until the code is pushed to the code, to monitor the application and system in the production. Automation in DevOps boosts speed, consistency, higher accuracy, reliability, and increases the number of deliveries.

24) What is the DevOps pipeline?

A pipeline in software engineering team is a set of automated processes which allows DevOps professionals and developer to reliably and efficiently compile, build, and deploy their code to their production compute platforms.

25) What is Azure DevOps?

Azure DevOps is also known as Microsoft visual studio team services (VSTS). It is a set of collaborative development tools built for the cloud. VSTS was commonly used as a standalone term, and Azure DevOps is a platform which is made up of a few different products, such as:

- Azure Test Plans
- Azure Boards
- Azure Repos
- Azure Pipeline
- Azure Artifacts

26) Which makes AWS DevOps highly accessible?

Here are some reasons which make AWS DevOps a highly popular, such as:

- AWS CloudFormation
- AWS EC2
- AWS CloudWatch
- AWS CodePipeline

27) What is the Build in DevOps?

The build is a method in which the source code is put together to check whether it works as a single unit. In the build creation process, the source code will undergo compilation, testing, inspection, and deployment.

28) Explain at what instance has used the SSH?

You can use SSH to log into a remote machine and work on the command line. Also, you can use it to tunnel into the system to facilitate secure encrypted communication between two hosts over an insecure network.

29) How infrastructure code is executed in AWS?

In AWS:

- The code for infrastructure will be in a simple JSON format.
- This JSON code will be organized into files called templates.
- These templates can be deployed on AWS DevOps and then managed as stacks.
- Later the CloudFormation service will do the Creating, deleting, updating, etc. operation in the stack.

30) Explain Git Bisect?

Git Bisect helps to find the commit which introduced a bug using binary search.

Module – 01

1. Define DevOps? List the different tools used in the DevOps.

- The DevOps is the combination of two words, one is Development and other is Operations. It is a culture to promote the development and operation process collectively.
- The DevOps tools such as Git, Ansible, Docker, Puppet, Jenkins, Chef, Nagios, and Kubernetes.

2. What is DevOps? Explain in Detail

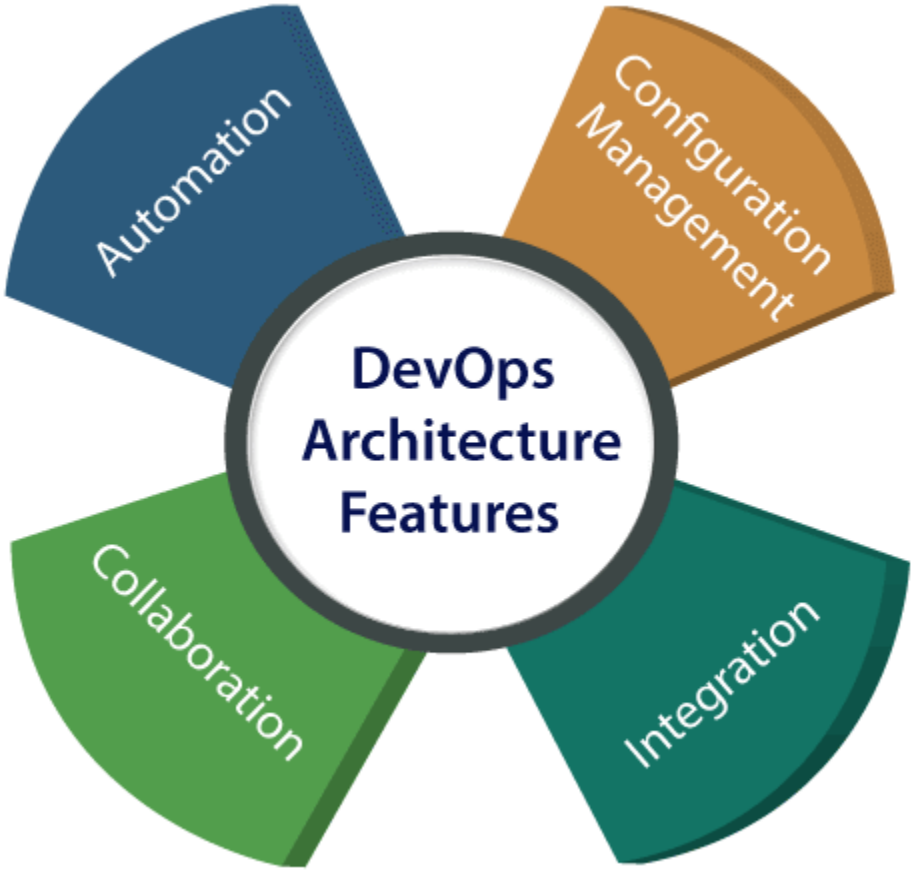
- The DevOps is a combination of two words, one is software Development, and second is Operations. This allows a single team to handle the entire application lifecycle, from development to testing, deployment, and operations. DevOps helps you to reduce the disconnection between software developers, quality assurance (QA) engineers, and system administrators.
- DevOps promotes collaboration between Development and Operations team to deploy code to production faster in an automated & repeatable way.
- DevOps helps to increase organization speed to deliver applications and services. It also allows organizations to serve their customers better and compete more strongly in the market.
- DevOps can also be defined as a sequence of development and IT operations with better communication and collaboration.
- DevOps has become one of the most valuable business disciplines for enterprises or organizations. With the help of DevOps, quality, and speed of the application delivery has improved to a great extent.
- DevOps is nothing but a practice or methodology of making "Developers" and "Operations" folks work together.
- DevOps represents a change in the IT culture with a complete focus on rapid IT service delivery through the adoption of agile practices in the context of a system-oriented approach.
- DevOps is all about the integration of the operations and development process.
- Organizations that have adopted DevOps noticed a 22% improvement in software quality and a 17% improvement in application deployment frequency and achieve a 22% hike in customer satisfaction. 19% of revenue hikes as a result of the successful DevOps implementation

3. Why DevOps explain

- Before going further, we need to understand why we need the DevOps over the other methods.
- The operation and development team worked in complete isolation.

- After the design-build, the testing and deployment are performed respectively. That's why they consumed more time than actual build cycles.
- Without the use of DevOps, the team members are spending a large amount of time on designing, testing, and deploying instead of building the project.
- Manual code deployment leads to human errors in production.
- Coding and operation teams have their separate timelines and are not in synch, causing further delays.

4. Explain DevOps Architecture features with diagram.



Automation

- Automation can reduce time consumption, especially during the testing and deployment phase. The productivity increases, and releases are made quicker by automation.
- This will lead in catching bugs quickly so that it can be fixed easily. For contiguous delivery, each code is defined through automated tests, cloudbased services, and builds. This promotes production using automated deploys.

Collaboration

The Development and Operations team collaborates as a DevOps team, which improves the cultural model as the teams become more productive with their productivity, which strengthens accountability and ownership. The teams share their responsibilities and work closely in sync, which in turn makes the deployment to production faster.

Integration

- Applications need to be integrated with other components in the environment.
- The integration phase is where the existing code is combined with new functionality and then tested.
- Continuous integration and testing enable continuous development. The frequency in the releases and micro-services leads to significant operational challenges. To overcome such problems, continuous integration and delivery are implemented to deliver in a **quicker, safer**, and **reliable manner**.

Configuration management

- It ensures the application to interact with only those resources that are concerned with the environment in which it runs.
- The configuration files are not created where the external configuration to the application is separated from the source code.
- The configuration file can be written during deployment, or they can be loaded at the run time, depending on the environment in which it is running.

5. List all the DevOps Advantages and Disadvantages.

Advantages

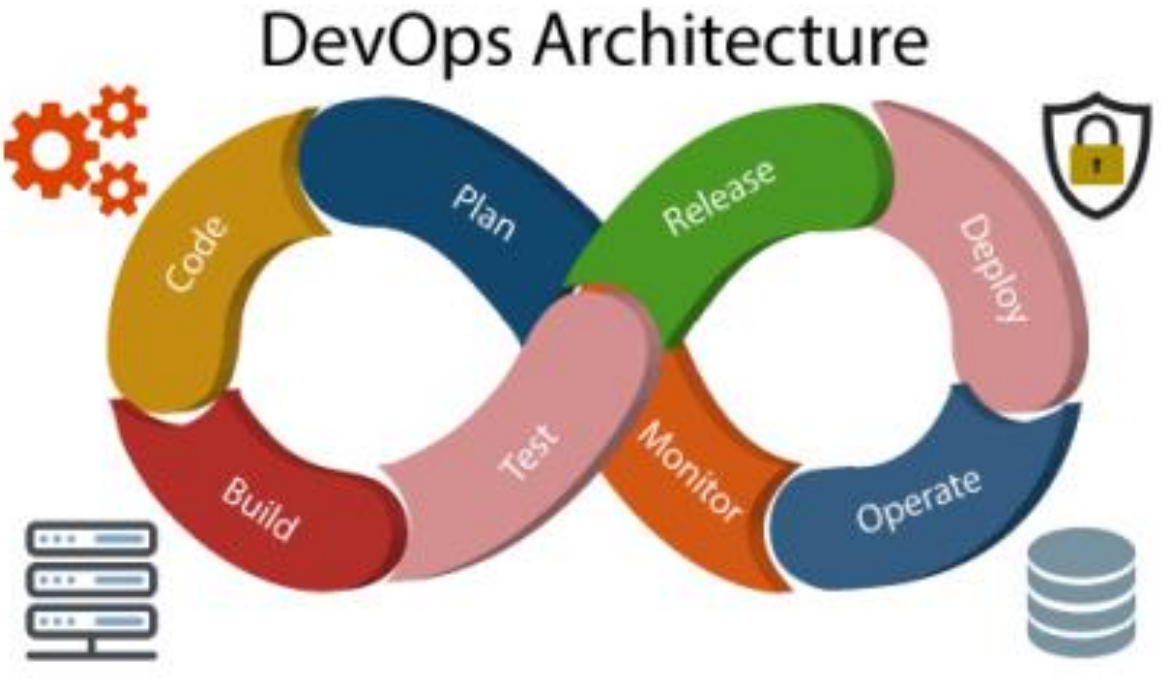
- DevOps is an excellent approach for quick development and deployment of applications.
- It responds faster to the market changes to improve business growth.
- DevOps escalate business profit by decreasing software delivery time and transportation costs.
- DevOps clears the descriptive process, which gives clarity on product development and delivery.
- It improves customer experience and satisfaction.
- DevOps simplifies collaboration and places all tools in the cloud for customers to access.
- DevOps means collective responsibility, which leads to better team engagement and productivity.

Disadvantages

- DevOps professional or expert's developers are less available.
- Developing with DevOps is so expensive.
- Adopting new DevOps technology into the industries is hard to manage in short time.
- Lack of DevOps knowledge can be a problem in the continuous integration of automation projects.

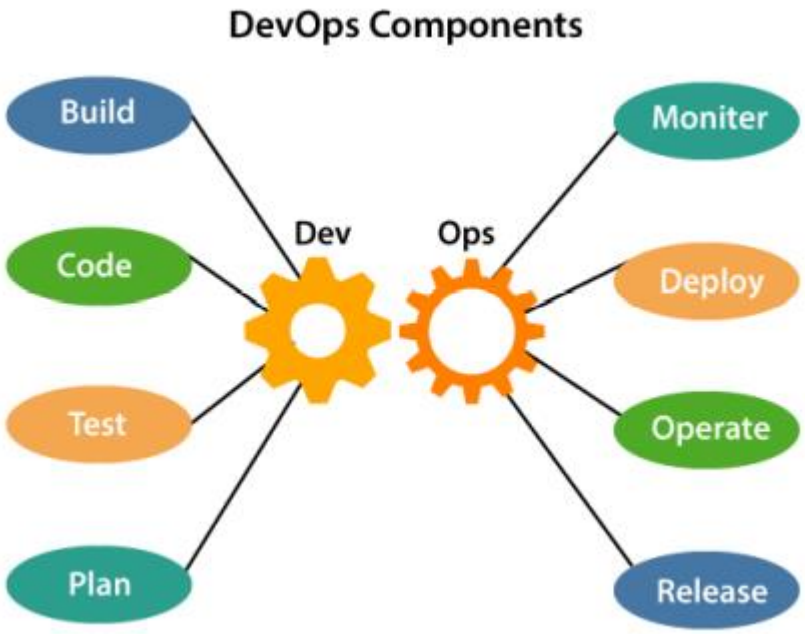
6. Explain DevOps Architecture with diagram.

DevOps Architecture



- Development and operations both play essential roles in order to deliver applications.
- The deployment comprises analysing the requirements, designing, developing, and testing of the software components or frameworks.
- The operation consists of the administrative processes, services, and support for the software.
- When both the development and operations are combined with collaborating, then the DevOps architecture is the solution to fix the gap between deployment and operation terms; therefore, delivery can be faster.
- DevOps architecture is used for the applications hosted on the cloud platform and large distributed applications.
- Agile Development is used in the DevOps architecture so that integration and delivery can be contiguous.
- When the development and operations team works separately from each other, then it is time-consuming to design, test, and deploy. And if the terms are not in sync with each other, then it may cause a delay in the delivery. So DevOps enables the teams to change their shortcomings and increases productivity.

7. Explain DevOps components with diagram and list and explain the DevOps components.



1) Build

Without DevOps, the cost of the consumption of the resources was evaluated based on the pre-defined individual usage with fixed hardware allocation. And with DevOps, the usage of cloud, sharing of resources comes into the picture, and the build is dependent upon the user's need, which is a mechanism to control the usage of resources or capacity.

2) Code

Many good practices such as Git enables the code to be used, which ensures writing the code for business, helps to track changes, getting notified about the reason behind the difference in the actual and the expected output, and if necessary reverting to the original code developed. The code can be appropriately arranged in files, folders, etc. And they can be reused.

3) Test

The application will be ready for production after testing. In the case of manual testing, it consumes more time in testing and moving the code to the output. The testing can be automated, which decreases the time for testing so that the time to deploy the code to production can be reduced as automating the running of the scripts will remove many manual steps.

4) Plan

DevOps use Agile methodology to plan the development. With the operations and development team in sync, it helps in organizing the work to plan accordingly to increase productivity.

5) Monitor

Continuous monitoring is used to identify any risk of failure. Also, it helps in tracking the system accurately so that the health of the application can be checked. The monitoring becomes more comfortable with services where the log data may get monitored through many third-party tools such as Splunk.

6) Deploy

Many systems can support the scheduler for automated deployment. The cloud management platform enables users to capture accurate insights and view the optimization scenario, analytics on trends by the deployment of dashboards.

7) Operate

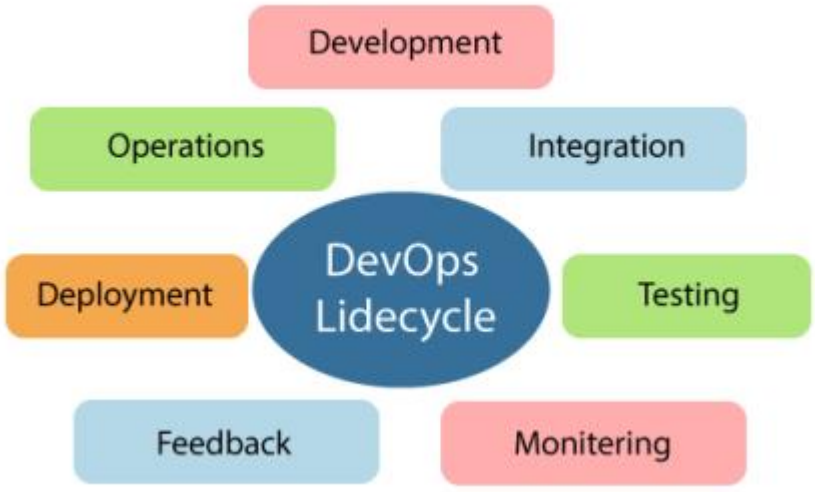
DevOps changes the way traditional approach of developing and testing separately. The teams operate in a collaborative way where both the teams actively participate throughout the service lifecycle. The operation team interacts with developers, and they come up with a monitoring plan which serves the IT and business requirements.

8) Release

Deployment to an environment can be done by automation. But when the deployment is made to the production environment, it is done by manual triggering. Many processes involved in release management commonly used to do the deployment in the production environment manually to lessen the impact on the customers.

8. Explain DevOps lifecycle with Diagram.

DevOps defines an agile relationship between operations and Development. It is a process that is practiced by the development team and operational engineers together from beginning to the final stage of the product.



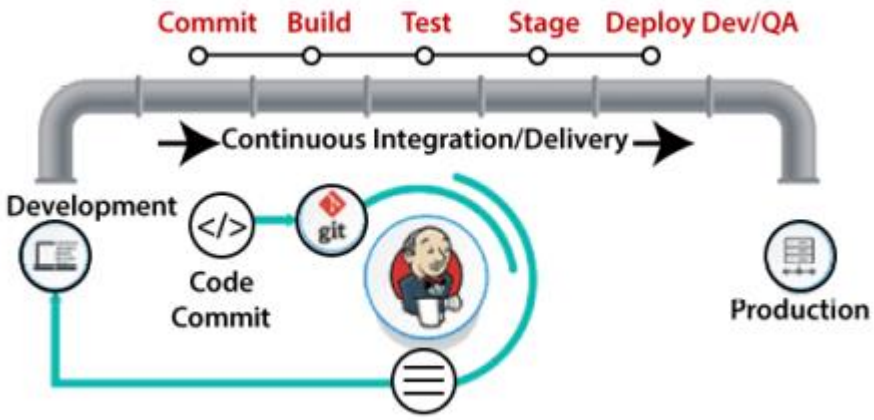
Learning DevOps is not complete without understanding the DevOps lifecycle phases. The DevOps lifecycle includes seven phases as given below:

1) Continuous Development

This phase involves the planning and coding of the software. The vision of the project is decided during the planning phase. And the developers begin developing the code for the application. There are no DevOps tools that are required for planning, but there are several tools for maintaining the code.

2) Continuous Integration

- This stage is the heart of the entire DevOps lifecycle. It is a software development practice in which the developers require to commit changes to the source code more frequently. This may be on a daily or weekly basis. Then every commit is built, and this allows early detection of problems if they are present. Building code is not only involved compilation, but it also includes unit testing, integration testing, code review, and packaging.
- The code supporting new functionality is continuously integrated with the existing code. Therefore, there is continuous development of software. The updated code needs to be integrated continuously and smoothly with the systems to reflect changes to the end-users.



- Jenkins is a popular tool used in this phase. Whenever there is a change in the Git repository, then Jenkins fetches the updated code and prepares a build of that code, which is an executable file in the form of war or jar. Then this build is forwarded to the test server or the production server.

3) Continuous Testing

- This phase, where the developed software is continuously testing for bugs. For constant testing, automation testing tools such as TestNG, JUnit, Selenium, etc are used. These tools allow QAs to test multiple code-bases thoroughly in parallel to ensure that there is no flaw in the functionality. In this phase, Docker Containers can be used for simulating the test environment.

- Selenium does the automation testing, and TestNG generates the reports. This entire testing phase can automate with the help of a Continuous Integration tool called Jenkins.
- Automation testing saves a lot of time and effort for executing the tests instead of doing this manually. Apart from that, report generation is a big plus. The task of evaluating the test cases

that failed in a test suite gets simpler. Also, we can schedule the execution of the test cases at predefined times. After testing, the code is continuously integrated with the existing code.

4) Continuous Monitoring

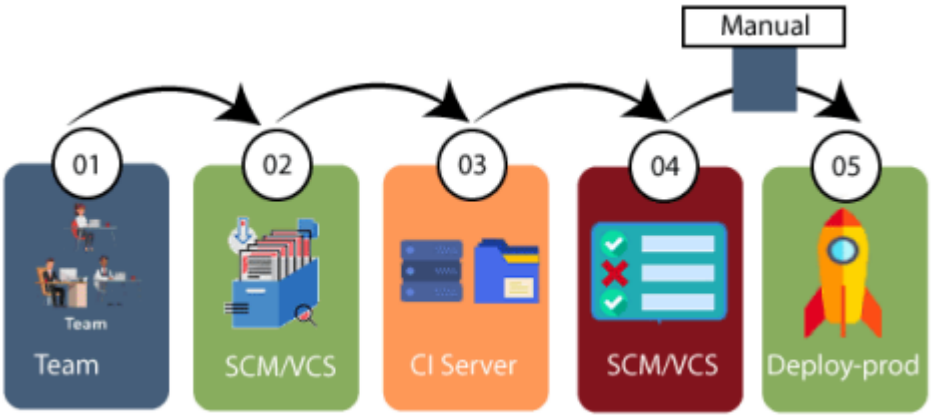
- Monitoring is a phase that involves all the operational factors of the entire DevOps process, where important information about the use of the software is recorded and carefully processed to find out trends and identify problem areas. Usually, the monitoring is integrated within the operational capabilities of the software application.
- It may occur in the form of documentation files or maybe produce largescale data about the application parameters when it is in a continuous use position. The system errors such as server not reachable, low memory, etc are resolved in this phase. It maintains the security and availability of the service.

5) Continuous Feedback

- The application development is consistently improved by analyzing the results from the operations of the software. This is carried out by placing the critical phase of constant feedback between the operations and the development of the next version of the current software application.
- The continuity is the essential factor in the DevOps as it removes the unnecessary steps which are required to take a software application from development, using it to find out its issues and then producing a better version. It kills the efficiency that may be possible with the app and reduce the number of interested customers.

6) Continuous Deployment

- In this phase, the code is deployed to the production servers. Also, it is essential to ensure that the code is correctly used on all the servers.



- The new code is deployed continuously, and configuration management tools play an essential role in executing tasks frequently and quickly. Here are some popular tools which are used in this phase, such as Chef, Puppet, Ansible, and SaltStack.
- Containerization tools are also playing an essential role in the deployment phase. Vagrant and Docker are popular tools that are used for this purpose. These tools help to produce

consistency across development, staging, testing, and production environment. They also help in scaling up and scaling down instances softly.

- Containerization tools help to maintain consistency across the environments where the application is tested, developed, and deployed. There is no chance of errors or failure in the production environment as they package and replicate the same dependencies and packages used in the testing, development, and staging environment. It makes the application easy to run on different computers.

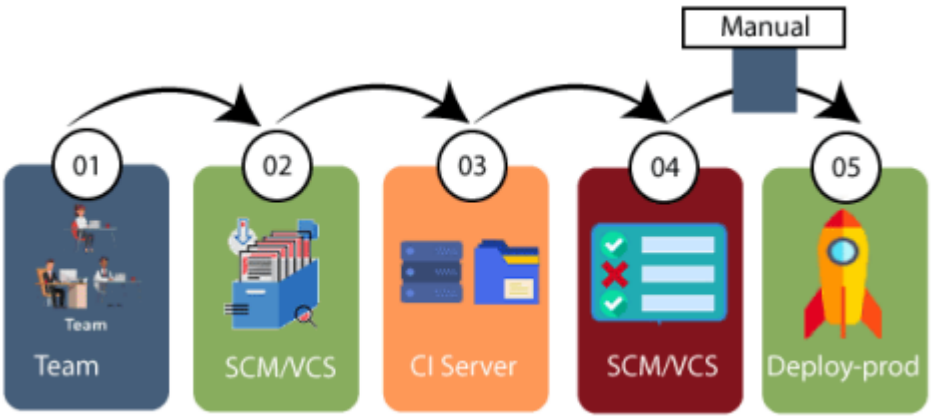
7) Continuous Operations

- All DevOps operations are based on the continuity with complete automation of the release process and allow the organization to accelerate the overall time to market continuingly.
- It is clear from the discussion that continuity is the critical factor in the DevOps in removing steps that often distract the development, take it longer to detect issues and produce a better version of the product after several months. With DevOps, we can make any software product more efficient and increase the overall count of interested customers in your product.

9. Explain continuous Deployment, Continuous Testing with diagram.

Continuous Deployment

- In this phase, the code is deployed to the production servers. Also, it is essential to ensure that the code is correctly used on all the servers.



- The new code is deployed continuously, and configuration management tools play an essential role in executing tasks frequently and quickly. Here are some popular tools which are used in this phase, such as Chef, Puppet, Ansible, and SaltStack.
- Containerization tools are also playing an essential role in the deployment phase. Vagrant and Docker are popular tools that are used for this purpose. These tools help to produce consistency across development, staging, testing, and production environment. They also help in scaling up and scaling down instances softly.
- Containerization tools help to maintain consistency across the environments where the application is tested, developed, and deployed. There is no chance of errors or failure in the

production environment as they package and replicate the same dependencies and packages used in the testing, development, and staging environment. It makes the application easy to run on different computers.

Continuous Testing

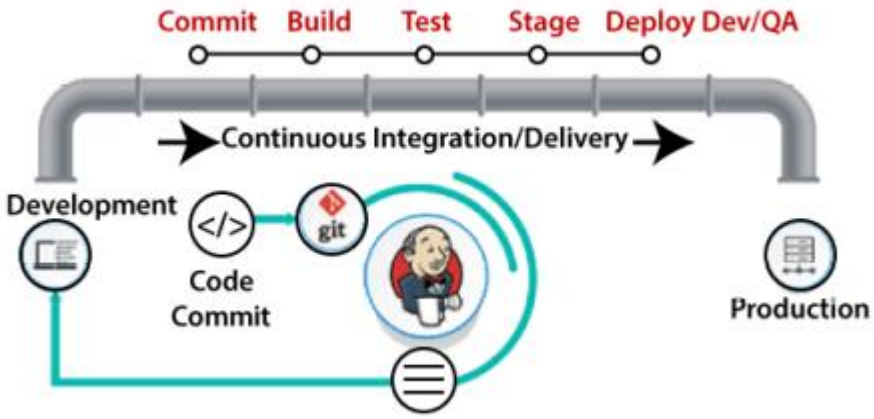
- This phase, where the developed software is continuously testing for bugs. For constant testing, automation testing tools such as TestNG, JUnit, Selenium, etc are used. These tools allow QAs to test multiple code-bases thoroughly in parallel to ensure that there is no flaw in the functionality. In this phase, Docker Containers can be used for simulating the test environment.

- Selenium does the automation testing, and TestNG generates the reports. This entire testing phase can automate with the help of a Continuous Integration tool called Jenkins.
- Automation testing saves a lot of time and effort for executing the tests instead of doing this manually. Apart from that, report generation is a big plus. The task of evaluating the test cases that failed in a test suite gets simpler. Also, we can schedule the execution of the test cases at predefined times. After testing, the code is continuously integrated with the existing code.

10. Explain continuous integration with Diagram.

Continuous Integration

- This stage is the heart of the entire DevOps lifecycle. It is a software development practice in which the developers require to commit changes to the source code more frequently. This may be on a daily or weekly basis. Then every commit is built, and this allows early detection of problems if they are present. Building code is not only involved compilation, but it also includes unit testing, integration testing, code review, and packaging.
- The code supporting new functionality is continuously integrated with the existing code. Therefore, there is continuous development of software. The updated code needs to be integrated continuously and smoothly with the systems to reflect changes to the end-users.



- Jenkins is a popular tool used in this phase. Whenever there is a change in the Git repository, then Jenkins fetches the updated code and prepares a build of that code, which is an executable file in the form of war or jar. Then this build is forwarded to the test server or the production server.

11. .Explain Automation and Configuration in DevOps Architecture features.

Automation

- Automation can reduce time consumption, especially during the testing and deployment phase. The productivity increases, and releases are made quicker by automation.
- This will lead in catching bugs quickly so that it can be fixed easily. For contiguous delivery, each code is defined through automated tests, cloudbased services, and builds. This promotes production using automated deploys.

Configuration management

- It ensures the application to interact with only those resources that are concerned with the environment in which it runs.
- The configuration files are not created where the external configuration to the application is separated from the source code.
- The configuration file can be written during deployment, or they can be loaded at the run time, depending on the environment in which it is running.

12. Explain Build, Code, and Test & Plan in Devops components.

1) Build

Without DevOps, the cost of the consumption of the resources was evaluated based on the pre-defined individual usage with fixed hardware allocation. And with DevOps, the usage of cloud, sharing of resources comes into the picture, and the build is dependent upon the user's need, which is a mechanism to control the usage of resources or capacity.

2) Code

Many good practices such as Git enables the code to be used, which ensures writing the code for business, helps to track changes, getting notified about the reason behind the difference in the actual and the expected output, and if necessary reverting to the original code developed. The code can be appropriately arranged in files, folders, etc. And they can be reused.

3) Test

The application will be ready for production after testing. In the case of manual testing, it consumes more time in testing and moving the code to the output. The testing can be automated, which decreases the time for testing so that the time to deploy the code to production can be reduced as automating the running of the scripts will remove many manual steps.

4) Plan

DevOps use Agile methodology to plan the development. With the operations and development team in sync, it helps in organizing the work to plan accordingly to increase productivity.

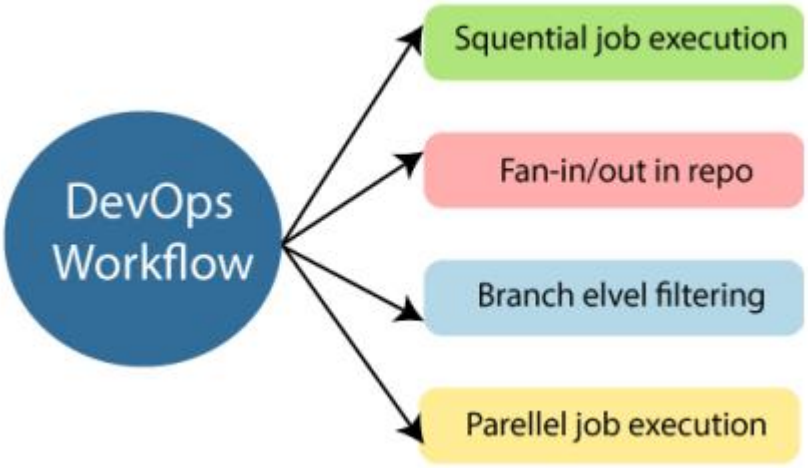
13. What is DevOps in Developers & Testers and IT operations?

- DevOps promotes collaboration between Development and Operations team to deploy code to production faster in an automated & repeatable way.
- DevOps helps to increase organization speed to deliver applications and services. It also allows organizations to serve their customers better and compete more strongly in the market.
- DevOps can also be defined as a sequence of development and IT operations with better communication and collaboration.
- DevOps has become one of the most valuable business disciplines for enterprises or organizations. With the help of DevOps, quality, and speed of the application delivery has improved to a great extent.
- DevOps is nothing but a practice or methodology of making "Developers" and "Operations" folks work together.
- DevOps represents a change in the IT culture with a complete focus on rapid IT service delivery through the adoption of agile practices in the context of a system-oriented approach.
- DevOps is all about the integration of the operations and development process.
- Organizations that have adopted DevOps noticed a 22% improvement in software quality and a 17% improvement in application deployment frequency and achieve a 22% hike in customer satisfaction. 19% of revenue hikes as a result of the successful DevOps implementation.

MODULE-2

1. Explain DevOps workflow with Diagram.

DevOps workflow provides a visual overview of the sequence in which input is provided. Also, it tells about which one action is performed, and output is generated for an operations process.



DevOps workflow allows the ability to separate and arrange the jobs which are top requested by the users. Also, it gives the ability to mirror their ideal process in the configuration jobs.

2. List and Explain different DevOps principles.

DevOps Principles

- The main principles of DevOps are Continuous delivery, automation, and fast reaction to the feedback.
- **End to End Responsibility:** DevOps team need to provide performance support until they become the end of life. It enhances the responsibility and the quality of the products engineered.
- **Continuous Improvement:** DevOps culture focuses on continuous improvement to minimize waste. It continuously speeds up the growth of products or services offered.
- **Automate Everything:** Automation is an essential principle of the DevOps process. This is for software development and also for the entire infrastructure landscape.
- **Custom Centric Action:** DevOps team must take customercentric for that they should continuously invest in products and services.
- **Monitor and test everything:** The DevOps team needs to have robust monitoring and testing procedures.
- **Work as one team:** In the DevOps culture role of the designers, developers, and testers are already defined. All they needed to do is work as one team with complete collaboration.
- These principles are achieved through several DevOps practices, which include frequent deployments, QA automation, and continuous delivery, validating ideas as early as possible and inteam collaboration

3. Explain end to end responsibility and continuous improvement and Automate Everything used in DevOps principles.

- **End to End Responsibility:** DevOps team need to provide performance support until they become the end of life. It enhances the responsibility and the quality of the products engineered.
- **Continuous Improvement:** DevOps culture focuses on continuous improvement to minimize waste. It continuously speeds up the growth of products or services offered.
- **Automate Everything:** Automation is an essential principle of the DevOps process. This is for software development and also for the entire infrastructure landscape.

4. List different DevOps practices identified in DevOps.

Some identified DevOps practices are:

1. Self-service configuration
2. Continuous build
3. Continuous integration
4. Continuous delivery
5. Incremental testing
6. Automated provisioning
7. Automated release management

5. List the popular DevOps tools used in DevOps.

- Puppet
- Ansible
- Docker
- Nagios
- Chef
- Jenkins
- Git
- SaltStack
- Splunk
- Selenium

6. Explain and List the features of different tools used in DevOps.

1) Puppet

- Puppet is the most widely used DevOps tool.

- It allows the delivery and release of the technology changes quickly and frequently. It has features of versioning, automated testing, and continuous delivery. It enables to manage entire infrastructure as code without expanding the size of the team.

Features

- Real-time context-aware reporting.
- Model and manage the entire environment.
- Defined and continually enforce infrastructure.
- Desired state conflict detection and remediation.
- It inspects and reports on packages running across the infrastructure.
- It eliminates manual work for the software delivery process.
- It helps the developer to deliver great software quickly.

2) Ansible

- Ansible is a leading DevOps tool.
- Ansible is an open-source IT engine that automates application deployment, cloud provisioning and other IT tools. It makes it easier for DevOps teams to scale automation and speed up productivity.
- Ansible is easy to deploy because it does not use any agents or custom security infrastructure on the client-side, and by pushing modules to the clients. These modules are executed locally on the client-side, and the output is pushed back to the Ansible server.

Features

- It is easy to use to open source deploy applications.
- It helps in avoiding complexity in the software development process.
- It eliminates repetitive tasks.
- It manages complex deployments and speeds up the development process.

3) Docker

- Docker is a high-end DevOps tool that allows building, ship, and run distributed applications on multiple systems. It also helps to assemble the apps quickly from the components, and it is typically suitable for container management.

Features

- It configures the system more comfortable and faster.
- It increases productivity.
- It provides containers that are used to run the application in an isolated environment.
- It routes the incoming request for published ports on available nodes to an active container. This feature enables the connection even if there is no task running on the node.

- It allows saving secrets into the swarm itself.

4) Nagios

- Nagios is one of the more useful tools for DevOps. It can determine the errors and rectify them with the help of network, infrastructure, server, and log monitoring systems.

Features

- It provides complete monitoring of desktop and server operating systems.
- The network analyser helps to identify bottlenecks and optimize bandwidth utilization.
- It helps to monitor components such as services, application, OS, and network protocol.
- It also provides to complete monitoring of Java Management Extensions.

5) CHEF

- A chef is a useful tool for achieving scale, speed, and consistency. The chef is a cloud-based system and open source technology. This technology uses Ruby encoding to develop essential building blocks such as recipes and cookbooks. The chef is used in infrastructure automation and helps in reducing manual and repetitive tasks for infrastructure management.
- Chef has got its convention for different building blocks, which are required to manage and automate infrastructure.

Features

- It maintains high availability.
- It can manage multiple cloud environments.
- It uses popular Ruby language to create a domain-specific language.
- The chef does not make any assumptions about the current status of the node. It uses its mechanism to get the current state of the machine.

6) Jenkins

- Jenkins is a DevOps tool for monitoring the execution of repeated tasks. Jenkins is a software that allows continuous integration. Jenkins will be installed on a server where the central build will take place. It helps to integrate project changes more efficiently by finding the issues quickly.

Features

- Jenkins increases the scale of automation.
- It can easily set up and configure via a web interface.
- It can distribute the tasks across multiple machines, thereby increasing concurrency.
- It supports continuous integration and continuous delivery.
- It offers 400 plugins to support the building and testing any project virtually.

- It requires little maintenance and has a built-in GUI tool for easy updates.

7) Git

- Git is an open-source distributed version control system that is freely available for everyone. It is designed to handle minor to major projects with speed and efficiency. It is developed to co- ordinate the work among programmers. The version control allows you to track and work together with your team members at the same workspace. It is used as a critical distributed version-control for the DevOps tool.

Features

- It is a free open source tool.
- It allows distributed development.
- It supports the pull request.
- It enables a faster release cycle.
- Git is very scalable.
- It is very secure and completes the tasks very fast.

8) SALTSTACK

- Stackify is a lightweight DevOps tool. It shows real-time error queries, logs, and more directly into the workstation. SALTSTACK is an ideal solution for intelligent orchestration for the software-defined data center.

Features

- It eliminates messy configuration or data changes.
- It can trace detail of all the types of the web request.
- It allows us to find and fix the bugs before production.
- It provides secure access and configures image caches.
- It secures multi-tenancy with granular role-based access control.
- Flexible image management with a private registry to store and manage images.

9) Splunk

- Splunk is a tool to make machine data usable, accessible, and valuable to everyone. It delivers operational intelligence to DevOps teams. It helps companies to be more secure, productive, and competitive.

Features

- It has the next-generation monitoring and analytics solution.
- It delivers a single, unified view of different IT services.
- Extend the Splunk platform with purpose-built solutions for security.

- Data drive analytics with actionable insight.

10) Selenium

- Selenium is a portable software testing framework for web applications. It provides an easy interface for developing automated tests.

Features

- It is a free open source tool.
- It supports multiplatform for testing, such as Android and ios.
- It is easy to build a keyword-driven framework for a WebDriver.
- It creates robust browser-based regression automation suites and tests.

7. Explain Puppet, Ansible and Docker DevOps tools used in DevOps and also List the features of mentioned DevOps tools.

1) Puppet

- Puppet is the most widely used DevOps tool.
- It allows the delivery and release of the technology changes quickly and frequently. It has features of versioning, automated testing, and continuous delivery. It enables to manage entire infrastructure as code without expanding the size of the team.

Features

- Real-time context-aware reporting.
- Model and manage the entire environment.
- Defined and continually enforce infrastructure.
- Desired state conflict detection and remediation.
- It inspects and reports on packages running across the infrastructure.
- It eliminates manual work for the software delivery process.
- It helps the developer to deliver great software quickly.

2) Ansible

- Ansible is a leading DevOps tool.
- Ansible is an open-source IT engine that automates application deployment, cloud provisioning and other IT tools. It makes it easier for DevOps teams to scale automation and speed up productivity.
- Ansible is easy to deploy because it does not use any agents or custom security infrastructure on the client-side, and by pushing modules to the clients. These modules are executed locally on the client-side, and the output is pushed back to the Ansible server.

Features

- It is easy to use to open source deploy applications.
- It helps in avoiding complexity in the software development process.
- It eliminates repetitive tasks.
- It manages complex deployments and speeds up the development process.

3) Docker

- Docker is a high-end DevOps tool that allows building, ship, and run distributed applications on multiple systems. It also helps to assemble the apps quickly from the components, and it is typically suitable for container management.

Features

- It configures the system more comfortable and faster.
- It increases productivity.
- It provides containers that are used to run the application in an isolated environment.
- It routes the incoming request for published ports on available nodes to an active container. This feature enables the connection even if there is no task running on the node.
- It allows saving secrets into the swarm itself.

8. Explain the DevOps tools Chef and Jenkins and List features of mentioned Tools.

CHEF

- A chef is a useful tool for achieving scale, speed, and consistency. The chef is a cloud-based system and open source technology. This technology uses Ruby encoding to develop essential building blocks such as recipes and cookbooks. The chef is used in infrastructure automation and helps in reducing manual and repetitive tasks for infrastructure management.
- Chef has got its convention for different building blocks, which are required to manage and automate infrastructure.

Features

- It maintains high availability.
- It can manage multiple cloud environments.
- It uses popular Ruby language to create a domain-specific language.
- The chef does not make any assumptions about the current status of the node. It uses its mechanism to get the current state of the machine.

Jenkins

- Jenkins is a DevOps tool for monitoring the execution of repeated tasks. Jenkins is a software that allows continuous integration. Jenkins will be installed on a server where the central build will take place. It helps to integrate project changes more efficiently by finding the issues quickly.

Features

- Jenkins increases the scale of automation.
- It can easily set up and configure via a web interface.
- It can distribute the tasks across multiple machines, thereby increasing concurrency.
- It supports continuous integration and continuous delivery.
- It offers 400 plugins to support the building and testing any project virtually.
- It requires little maintenance and has a built-in GUI tool for easy updates.

9. Explain and list the features of DevOps tools like Git and Saltstack from DevOps.

Git

- Git is an open-source distributed version control system that is freely available for everyone. It is designed to handle minor to major projects with speed and efficiency. It is developed to co- ordinate the work among programmers. The version control allows you to track and work together with your team members at the same workspace. It is used as a critical distributed version-control for the DevOps tool.

Features

- It is a free open source tool.
- It allows distributed development.
- It supports the pull request.
- It enables a faster release cycle.
- Git is very scalable.
- It is very secure and completes the tasks very fast.

SALTSTACK

- Stackify is a lightweight DevOps tool. It shows real-time error queries, logs, and more directly into the workstation. SALTSTACK is an ideal solution for intelligent orchestration for the software-defined data center.

Features

- It eliminates messy configuration or data changes.
- It can trace detail of all the types of the web request.
- It allows us to find and fix the bugs before production.
- It provides secure access and configures image caches.
- It secures multi-tenancy with granular role-based access control.
- Flexible image management with a private registry to store and manage images.

10. Explain DevOps Automation in Detail.

- Automation is the crucial need for DevOps practices, and automate everything is the fundamental principle of DevOps. Automation kick starts from the code generation on the developer's machine, until the code is pushed to the code and after that to monitor the application and system in the production.
- Automating infrastructure set up and configurations, and software deployment is the key highlight of DevOps practice. DevOps practice id is dependent on automation to make deliveries over a few hours and make frequent deliveries across platforms.
- Automation in DevOps boosts speed, consistency, higher accuracy, reliability, and increases the number of deliveries. Automation in DevOps encapsulates everything right from the building, deploying, and monitoring

11. Explain and list the features of splunk and selenium tools from DevOps.

Splunk

- Splunk is a tool to make machine data usable, accessible, and valuable to everyone. It delivers operational intelligence to DevOps teams. It helps companies to be more secure, productive, and competitive.

Features

- It has the next-generation monitoring and analytics solution.
- It delivers a single, unified view of different IT services.
- Extend the Splunk platform with purpose-built solutions for security.
- Data drive analytics with actionable insight.

Selenium

- Selenium is a portable software testing framework for web applications. It provides an easy interface for developing automated tests.

Features

- It is a free open source tool.
- It supports multiplatform for testing, such as Android and ios.
- It is easy to build a keyword-driven framework for a WebDriver.
- It creates robust browser-based regression automation suites and tests.

12. List the different categories classified in DevOps and explain each category in detail.

In large DevOps team that maintain extensive massive IT infrastructure can be classified into six categories, such as:

DevOps Automation Tools:

- Infrastructure Automation
- Configuration Management
- Deployment Automation
- Performance Management
- Log management
- Monitoring

13. Explain infrastructure Automation and also Explain the tools used in infrastructure automation.

DevOps Automation

Automation is the crucial need for DevOps practices, and automate everything is the fundamental principle of DevOps. Automation kick starts from the code generation on the developers machine, until the code is pushed to the code and after that to monitor the application and system in the production.

DevOps Automation Tools:

- Infrastructure Automation
- Configuration Management
- Deployment Automation
- Performance Management
- Log management
- Monitoring

Infrastructure Automation

Amazon Web Services (AWS): Being a cloud service, you don't need to be physically present in the data center, they are easy to scale ondemand, and there are no up-front hardware costs. It can be configured to provide more servers based on traffic automatically.

Configuration Management

Chef: Chef is a handy DevOps tool for achieving speed, scale, and consistency. It can be used to ease out of complex tasks and perform configuration management. With the help of this tool, the DevOps team can avoid making changes across ten thousand servers. Rather, they need to make changes in one place, which is automatically reflected in other servers.

Deployment Automation

Jenkins: It facilitates continuous integration and testing. It helps to integrate project changes more efficiently by quickly finding issues as soon as built is deployed.

Performance Management

App Dynamic: It offers real-time performance monitoring. The data collected by this tool help developers to debug when issues occur.

Log Management

Splunk: This DevOps tool solves issues such as storing, aggregating, and analyzing all logs in one place.

Monitoring

Nagios: It notified people when infrastructure and related service go down. Nagios is a tool for this purpose, which helps the DevOps team to find and correct problems

14. Explain configuration Management and Deployment Automation in DevOps tools.

Configuration Management

Chef: Chef is a handy DevOps tool for achieving speed, scale, and consistency. It can be used to ease out of complex tasks and perform configuration management. With the help of this tool, the DevOps team can avoid making changes across ten thousand servers. Rather, they need to make changes in one place, which is automatically reflected in other servers.

Deployment Automation

Jenkins: It facilitates continuous integration and testing. It helps to integrate project changes more efficiently by quickly finding issues as soon as built is deployed

15. Explain performance and Log Management and Monitoring Automation tools used in DevOps

Log Management

Splunk: This DevOps tool solves issues such as storing, aggregating, and analyzing all logs in one place.

Monitoring

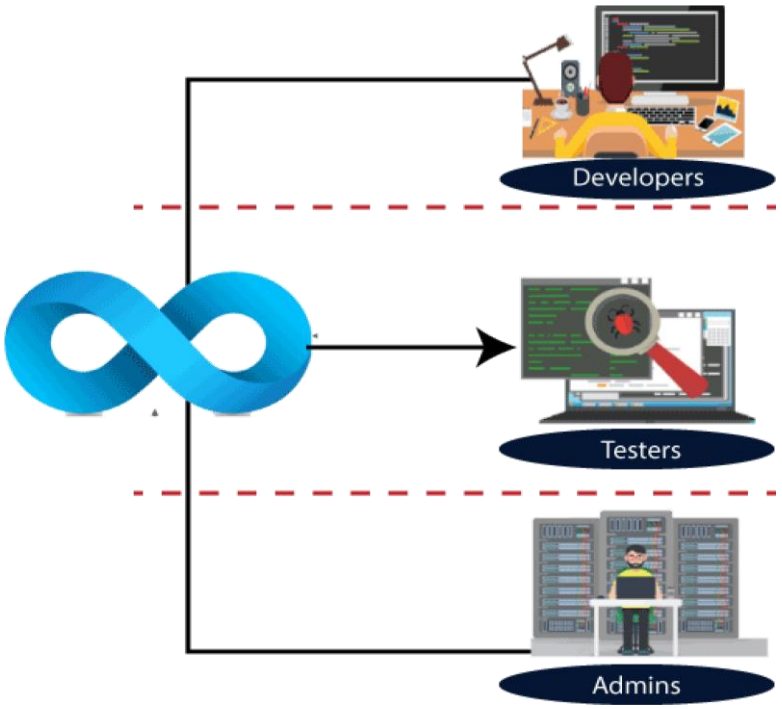
Nagios: It notified people when infrastructure and related service go down. Nagios is a tool for this purpose, which helps the DevOps team to find and correct problems.

Module – 03

1. Explain DevOps engineer with diagram?

DevOps Engineers

- DevOps Engineer is an IT professional who works with system operators, software developers, and other production IT staff to administer code releases.
- DevOps engineer understands the software development lifecycle and various automation tools for developing digital pipelines.
- DevOps have hard as well as soft skills to communicate and collaborate with development, testing, and operations teams.



- DevOps engineers need to code occasionally from scratch, and they must have the basics of software development languages.
- The DevOps engineer will work with development team staff to tackle the coding and scripting needed to connect elements of code, like libraries or software development kits.
- A bachelor's degree in computer science or related fields is generally required for DevOps engineers. Many companies prefer those who have a master's degree and at least three to five years of work experience in this field. HTTP, HTML, CSS, SSL, XML, Linux, Java, Amazon Web Services (AWS), NoSQL technologies, DNS, and web app development.

2. Explain different DevOps engineer rules and responsibilities?

DevOps engineers work full time. They are responsible for the production and continuing maintenance of a software application platform.

Below are some roles, responsibilities, and skills which are expected from DevOps engineers, such as:

- Manage projects effectively through an open standard based platform.
- Increases project visibility through traceability.
- Improve quality and reduce the development cost with collaboration.
- DevOps should have the soft skill of problem solver and a quick learner.
- Analyze, design, and evaluate automation scripts and systems.
- Able to perform system troubleshooting and problem-solving across the platform and application domains.
    - Ensuring the critical resolution of system issues by using the best cloud security solution services.

3. Explain DevOps pipline?

- A pipeline in software engineering team is a set of automated processes which allows DevOps professionals and developer to reliably and efficiently compile, build, and deploy their code to their production compute platforms.
- The most common components of a pipeline in DevOps are build automation or continuous integration, test automation, and deployment automation.

4. List the different DevOps pipline tools.

- Source control
- Build tools
- Containerization
- Configuration management
- Monitoring

5. Explain continues integration pipline and list semi significant benefit of CI.

Continuous integration (CI) is a practice in which developers can check their code into a version-controlled repository several times per day. Automated build pipelines are triggered by these checks which allows fast and easy to locate error detection.

Some significant benefits of CI are:

- Small changes are easy to integrate into large codebases.
- More comfortable for other team members to see what you have been working.
- Fewer integration issues allowing rapid code delivery.
- Bugs are identified early, making them easier to fix, resulting in less debugging work.

6. Explain continues delivery pipeline and list some significant length of the CD

Continuous delivery (CD) is the process that allows operation engineers and developers to deliver bug fixes, features, and configuration change into production reliably, quickly, and sustainably. Continuous delivery offers the benefits of code delivery pipelines, which are carried out that can be performed on demand.

Some significant benefits of the CD are:

- Faster bug fixes and features delivery.
- CD allows the team to work on features and bug fixes in small batches, which means user feedback received much quicker. It reduces the overall time and cost of the project.

7. Explain DevOps methodology

We have a demonstrated methodology that takes an approach to cloud adoption. It accounts for all the factors required for successful approval such as people, process, and technology, resulting in a focus on the following critical consideration:

- **The Teams:** Mission or project and cloud management.
- **Connectivity:** Public, on-premise, and hybrid cloud network access.
- **Automation:** Infrastructure as code, scripting the orchestration and deployment of resources.
- **On-boarding Process:** How the project gets started in the cloud.
- **Project Environment:** TEST, DEV, PROD (identical deployment, testing, and production).
- **Shared Services:** Common capabilities provided by the enterprise.
- **Naming Conventions:** Vital aspect to track resource utilization and billing.
- **Defining Standards Role across the Teams:** Permissions to access resources by job function.

8. Explain AWS in DevOps and list the different combination of AWS and explain briefly

AWS is the best cloud service provider, and DevOps is the implementation of the software development lifecycle.

Here are some reasons which make AWS DevOps a highly popular combination, such as:

- AWS CloudFormation
- AWS EC2
- AWS CloudWatch
- AWS CodePipeline

Let's see all of these by one in brief such as:

AWS CloudFormation

DevOps team is required to create and release cloud instances and services more frequently in comparison to development teams. Templates of AWS resources such as EC2 instances, ECS containers, and S3 storage buckets let you set up the entire stack without having to bring everything together.

AWS EC2

You can run containers inside EC2 instances. Hence you can leverage the AWS security and management features.

AWS CloudWatch

This monitoring tool tracks every resource that AWS has to offer. It makes it easy to use third-party tools for monitoring such as sumo logic etc.

AWS CodePipeline

- Code Pipeline is an essential feature from AWS, which highly simplifies the way you manage your CI/CD toolset. It integrates with tools such as **Jenkins, GitHub**, and Code Deploy that enable you to visually control the flow of app updates from build to production.

9. Explain Azure DevOps

- Azure DevOps is also known as Microsoft visual studio team services (VSTS). It is a set of collaborative development tools built for the cloud.
- VSTS was commonly used as a standalone term, and Azure DevOps is a platform which is made up of a few different products, such as:
    - Azure Test Plans
    - Azure Boards
    - Azure Repos
    - Azure Pipeline
        - Azure Artifacts
    - Azure DevOps is everything that needs to turn an idea into a working piece software. You can plan a project with azure tools.
    - The azure pipeline is the CI component of azure DevOps.
    - The azure pipeline is Microsoft's cloud-native continuous integration server, which allows teams to continuously build, test, and deploy all from the cloud.
    - An azure pipeline can connect to any number of source code repositories such as Azure Repos, GitHub, Tests, to grab code and artifacts for application delivery.

10. Explain Azure DevOps server and DevOps services

- Azure DevOps Server is a Microsoft product that provides version control, requirements management, reporting, lab management, project management, testing, automated builds,

and release management capabilities. It covers the entire application of lifecycle and enables DevOps capabilities.

- Azure DevOps can be used as a back-end to the numerous integrated development environments, but it is modified for Microsoft visual studio and eclipse on all platforms.
- Microsoft announced the release of the software as a service offering of visual studio on the Microsoft Azure platform at the time Microsoft called it a visual studio online.
- Microsoft offers visual studio, basic, and stakeholder subscriber access levels for the Azure DevOps services. The basic plan is free of cost for up to five users. Users with a visual studio subscription can be added to a project with no additional charge.

Module – 04

1. Explain DevOps training certification.

DevOps training certification helps anyone to make a career as a DevOps engineer. DevOps certifications are available from Red Hat, Amazon web services, DevOps institution, and Microsoft academy.

Let's see all of these certifications one by one in brief such as:

Red Hat Certification

Red Hat offers a different level of certifications for DevOps professional as follows:

- Red Hat certificate of expertise in the Ansible automation.
- Red Hat certificate of expertise in Platform-as-a-service.
- Red Hat certificate of expertise in Container Administrator.
- Red Hat certificate of expertise in Configuration Management.
- Red Hat certificate of expertise in the Containerized Application Development.

Amazon Web Service Certification

This certificate tests you on how to use the most common DevOps patterns to develop, deploy, and maintain applications on AWS. It also evaluates you on the core principle of the DevOps methodology.

Amazon web service certificate has two requisites. First, the certification fee is $300, and the second time duration is 170 minutes.

2. List the different certificates in DevOps and explain.

Let's see all of these certifications one by one in brief such as:

Red Hat Certification

Red Hat offers a different level of certifications for DevOps professional as follows:

- Red Hat certificate of expertise in the Ansible automation.
- Red Hat certificate of expertise in Platform-as-a-service.
- Red Hat certificate of expertise in Container Administrator.
- Red Hat certificate of expertise in Configuration Management.
- Red Hat certificate of expertise in the Containerized Application Development.

Amazon Web Service Certification

This certificate tests you on how to use the most common DevOps patterns to develop, deploy, and maintain applications on AWS. It also evaluates you on the core principle of the DevOps methodology.

Amazon web service certificate has two requisites. First, the certification fee is $300, and the second time duration is 170 minutes.

3. Explain Red Hat certification and Amazon Web Service certification.

Red Hat Certification

Red Hat offers a different level of certifications for DevOps professional as follows:

- Red Hat certificate of expertise in the Ansible automation.
- Red Hat certificate of expertise in Platform-as-a-service.
- Red Hat certificate of expertise in Container Administrator.
- Red Hat certificate of expertise in Configuration Management.
- Red Hat certificate of expertise in the Containerized Application Development.

Amazon Web Service Certification

This certificate tests you on how to use the most common DevOps patterns to develop, deploy, and maintain applications on AWS. It also evaluates you on the core principle of the DevOps methodology.

Amazon web service certificate has two requisites. First, the certification fee is $300, and the second time duration is 170 minutes.

4. Explain DevOps Institutions.

The DevOps institution is a global learning community around emerging DevOps practice. This organization is setting the quality standards for DevOps competency-based qualifications. Some offered certification courses are:

- DevOps Leader
- DevOps Test Engineer
- DevOps Foundation Certified
- DevOps Foundation
- Certified Agile Process Owner
- Certified Agile Service Manager
- Continuous Delivery Architecture
    - Develops Engineer

5. Explain DevOps vs Agile.

DevOps

The DevOps is a combination of two words, one is software Development, and second is Operations. This allows a single team to handle the entire application lifecycle, from development to **testing, deployment**, and **operations**. DevOps helps you to reduce the disconnection between software developers, quality assurance (QA) engineers, and system administrators.

- DevOps promotes collaboration between Development and Operations team to deploy code to production faster in an automated & repeatable way.
- DevOps helps to increase organization speed to deliver applications and services. It also allows organizations to serve their customers better and compete more strongly in the market.
- DevOps can also be defined as a sequence of development and IT operations with better communication and collaboration.
- DevOps has become one of the most valuable business disciplines for enterprises or organizations. With the help of DevOps, **quality**, and **speed** of the application delivery has improved to a great extent.
- DevOps is nothing but a practice or methodology of making "**Developers**" and "**Operations**" folks work together. DevOps represents a change in the IT culture with a complete focus on rapid IT service delivery through the adoption of agile practices in the context of a system-oriented approach.

Agile

The Agile involves continuous iteration of development and testing in the **SDLC** process. Both development and testing activities are concurrent, unlike the waterfall model. This software development method emphasizes on incremental, iterative, and evolutionary development.
It breaks the product into small pieces and integrates them for final testing. It can be implemented in many ways, such as **Kanban, XP, Scrum**, etc.
The Agile software development focus on the four core values, such as:

- Working software over comprehensive documentation.
- Responded to change over following a plan.
- Customer collaboration over contract negotiation.

Individual and team interaction over the process and tools.

6. What is Agile and explain Agile.

The Agile involves continuous iteration of development and testing in the **SDLC** process. Both development and testing activities are concurrent, unlike the waterfall model. This software development method emphasizes on incremental, iterative, and evolutionary development.
It breaks the product into small pieces and integrates them for final testing. It can be implemented in many ways, such as **Kanban, XP, Scrum**, etc.
The Agile software development focus on the four core values, such as:

- Working software over comprehensive documentation.
- Responded to change over following a plan.
- Customer collaboration over contract negotiation.

- Individual and team interaction over the process and tools.

7. List the different parameter in DevOps and explain each parameter.

or

8. List the different parameter in Agile and explain each parameter.

or

9. Explain some of the essential difference between the DevOps and Agile

Parameter	DevOps	Agile
Definition	DevOps is a practice of bringing development and operation teams together.	Agile refers to the continuous iterative approach, which focuses on collaboration, customer feedback, small, and rapid releases.
Purpose	DevOps purpose is to manage end to end engineering processes.	The agile purpose is to manage complex projects.
Task	It focuses on constant testing and delivery.	It focuses on constant changes.
Team size	It has a large team size as it involves all the stack holders.	It has a small team size. As smaller is the team, the fewer people work on it so that they can move faster.
Team skillset	The DevOps divides and spreads the skill set between development and the operation team.	The Agile development emphasizes training all team members to have a wide variety of similar and equal skills.
Implementation	DevOps is focused on collaboration, so it does not have any commonly accepted framework.	Agile can implement within a range of tactical frameworks such as **safe, scrum**, and **sprint**.
Duration	The ideal goal is to deliver the code to production daily or every few hours.	Agile development is managed in units of sprints. So this time is much less than a month for each sprint.
Target areas	End to End business solution and fast delivery.	Software development.

Feedback	Feedback comes from the internal team.	In Agile, feedback is coming from the customer.
Shift left principle	It supports both variations left and right.	It supports only shift left.
Focus	DevOps focuses on operational and business readiness.	Agile focuses on functional and non-functional readiness.
Importance	In DevOps, developing, testing, and implementation all are equally important.	Developing software is inherent to Agile.
Quality	DevOps contributes to creating better quality with automation and early bug removal. Developers need to follow Coding and best Architectural practices to maintain quality standards.	The Agile produces better applications suites with the desired requirements. It can quickly adapt according to the changes made on time during the project life.
Tools	**Puppet, Chef, AWS, Ansible**, and team City OpenStack are popular DevOps tools.	**Bugzilla, Kanboard, JIRA** are some popular Agile tools.
Automation	Automation is the primary goal of DevOps. It works on the principle of maximizing efficiency when deploying software.	Agile does not emphasize on the automation.
Communication	DevOps communication involves specs and design documents. It is essential for the operational team to fully understand the software release and its network implications for the enough running the deployment process.	Scrum is the most common method of implementing Agile software development. Scrum meeting is carried out daily.

Documentation	In the DevOps, the process documentation is foremost because it will send the software to an operational team for deployment. Automation minimizes the impact of insufficient documentation. However, in the development of sophisticated software, it's difficult to transfer all the knowledge required.	The agile method gives priority to the working system over complete documentation. It is ideal when you are flexible and responsive. However, it can harm when you are trying to turn things over to another team for deployment.